Scribe Notes: September 20, 2010

Compiled by Abhay Goyal

The class started with chocolate and an improved recall of names.

James was randomly picked again to solve the puzzle. He showed that more than 1 robot must be faulty.

Adeline was picked to show which ones were faulty; she showed C and B had to be the faulty robots.

We then discussed the possibilities if 3 robots could be faulty.

Reviewing last week’s lesson, we started with the Finite State Automaton representation of noun phrases.

Chomsky showed that this representation was not enough using “if” sentences. If “s” represents a sentence, he started with the basic if statement: if *s* then *s* else *s*. From this, he used nested if statements to make sentences with more “if”s—if *s* then if *s* then *s* else *s* else *s.*

Potentially, one could make an arbitrarily large number of “if”s in a single sentence, and although it would be a weird sentence, it would technically be grammatically correct. This would overwhelm the finite state automaton as, by definition, it is finite. (It’s not that infinite sentences itself overwhelm the finite state automaton, because noun phrases can be of unbounded length after all. It’s that these nested if/then/else sentences have to have as many ifs as thens and elses, so whichever representation is used must be able to encode a count of an arbitrarily large number. If, say, each state represents some number, then any finite state automaton must at some point be incapable of representing some number that is large enough.)\_ As such, the finite state automaton is an inadequate representation for at least if/then/else sentences in English.

John Bachus: invents Fortran programming language. He, while working for IBM, realized that programming in machine language (which we will discuss) was very difficult, so he decided to make a language that was easier to work with. Fortran was fast, but Bachus thought there could be a better language.

Another European group creates Algol: a new programming language. Bachus likes Algol more than Fortran. He thinks to go back to Chomsky for language description, and he uses the idea of context free grammar.

**Context-Free Grammar:**

 S: N V O | V O | if S then S else S

 N: it | Adeline | Simon | …

 V: take | is | …

 O: windbreaker | umbrella | …

Key: S = sentence, N = noun, V = verb, O = object, | = or

The basic idea of context-free grammar is that, well, grammar exists free of context. Simply put, these replacements do not rely on what’s surrounding them. You can do any of these replacements at any time and it should be grammatically correct.

A problem with this system is that it doesn’t work well with passive sentences, so is inadequate to natural language, but it is widely used for (people-invented) computer languages..

Next we tried applying the Context-Free model to noun phrases:

 NP: det N | det AdjList N

 AdjList: Adj Adjlist | Adj

[Abhay, please give an example parse tree. Also use pdf]